Deep learning lab

Computer vision track

Submitted by: Youssef Nassar 5165986

Task 1. Self-supervised Pre-training:

The goal of the first task is to train a feature representation using a self-supervised learning method (DINO). We are using the COCO unlabeled dataset, which we downloaded from the given link and added it to the tfpool. As mentioned in the paper, the teacher does not update the weights (no backpropagation) so we set its parameters require grad to false. Then we implement the train function where we follow the pseudo code available in the DINO paper. For the teacher model we use only the 2 global crops, but for the student model we use all of the crops (global and local). In the loss function initialisation in the train function we set the train parameter to True so that the that the teacher output can update its center. In the validate function we use the TSNE embedding to reduce the dimensionality of the features learned from 512 to only 2 so that we can be able to visualize it.

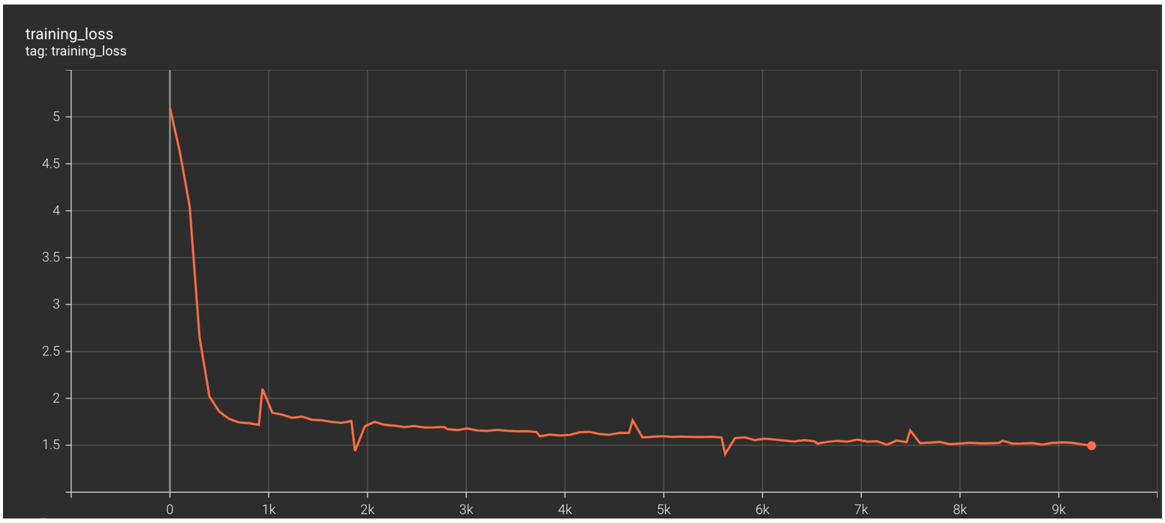


Figure 1: training loss over the steps

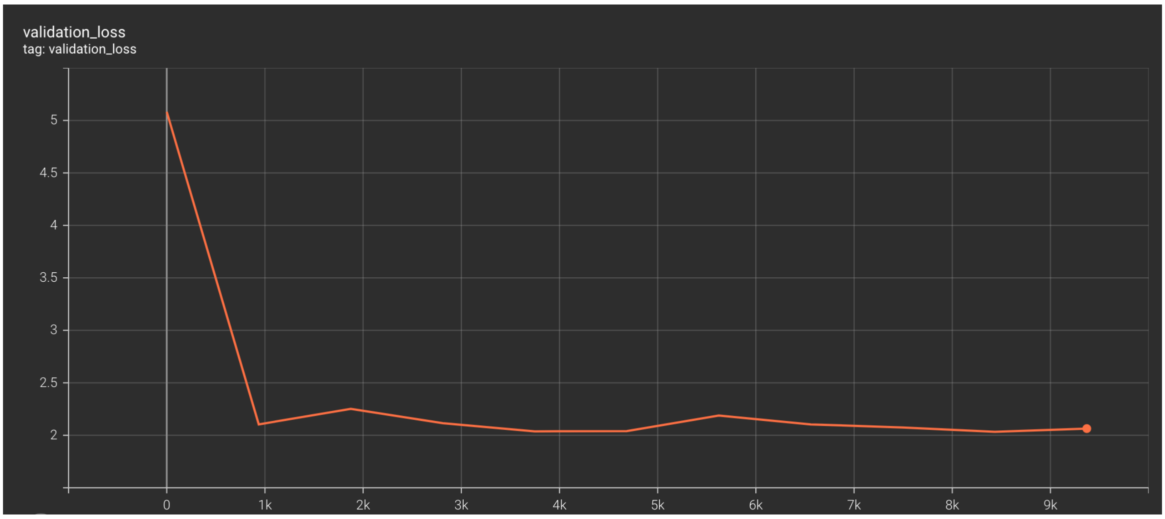
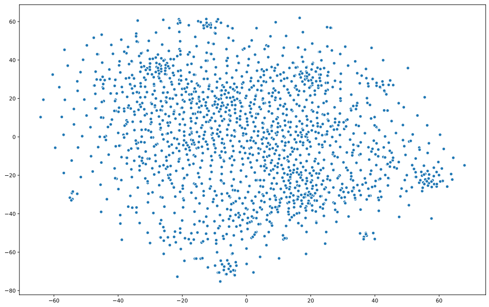
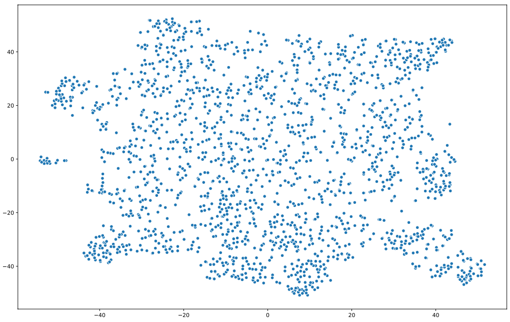


Figure 2: validation loss over the steps

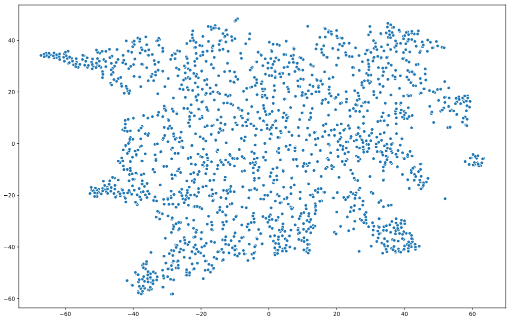
As we can see in the above figures that the training loss goes to almost 1.5 and the validation loss goes to almost 2, which means it's not overfitting but the performance is not that good, this could be to the fact that I reduced the batch size to 64 instead of 128 because when trying to run it with 128 on the tfpool the CUDA was out of memory.



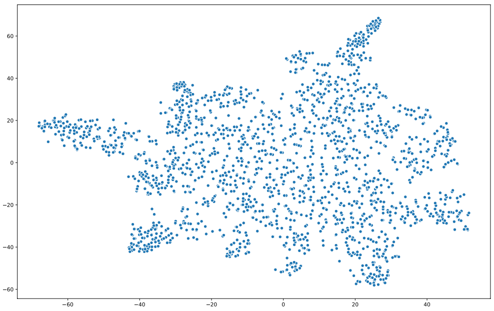
Epoch -1



Epoch 0

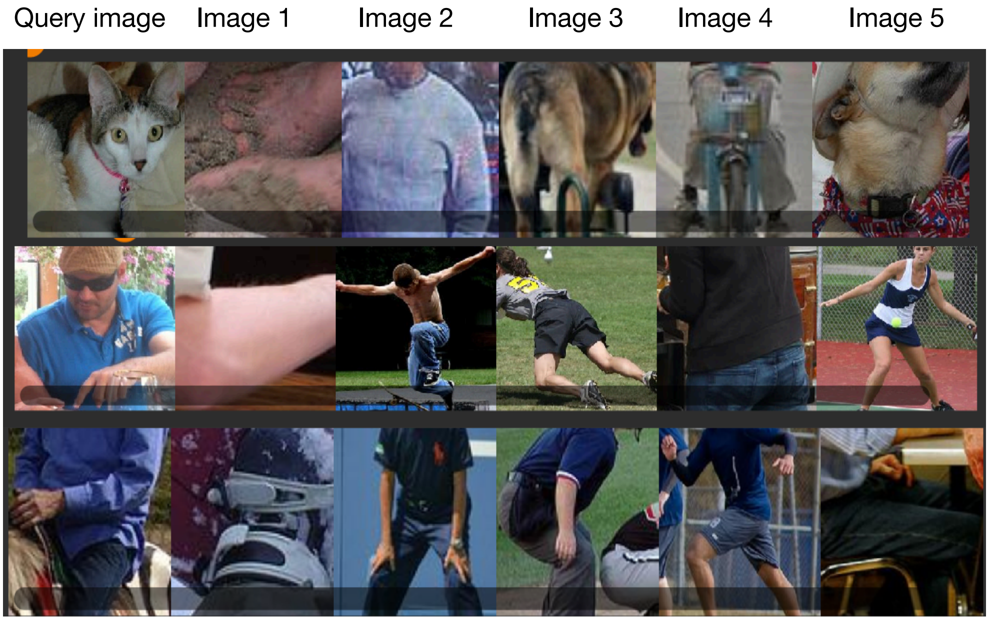


Epoch 5



Epoch 9

As we can see in the above graphs these are the output for the TSNE embedding for 4 different epochs. We can see throughout the epochs that the similar features start to form clusters, although there are some that are in somehow not clustered together, but this could be that TSNE loses a lot of information while embedding.



As we can see in the above table of images our model that we used (model of 9th epoch) to get the nearest neighbours's features learned some of the features but it is not working as good as we hoped. For example in the first query image it has a cat but it got some of the NN images with features that is not related to it, but also it gave 2 dogs images which can share some of the features with the cat. But for the second image it was a little better as it gave all humans with showing arms or legs. So we can conclude that the model we have is working fine but not the best performance though.